CASCADE ERROR PROJECTION _A LEARNING ALGORITHM FOR HARDWARE IMPLEMENTATION

Tuan A. Duong and Taher Daud
Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109

Abstract:

In this paper, we workout a detailed mathematical analysis for a new learning algorithm termed Cascade Error Projection (CEP) and a general learning frame work. This frame work can be used to obtain the cascade correlation learning algorithm by choosing a particular set of parameters. Furthermore, CEP learning algorithm is operated only on one layer, whereas the other set of weights can be calculated deterministically. In association with the dynamical stepsize change concept to convert the weight update from infinite space into a finite space, the relation between the current stepsize and the previous energy level is also given and the estimation procedure for optimal stepsize is used for validation of our proposed technique.

The weight values of zero are used for starting the learning for every layer, and a single hidden unit is applied instead of using a pool of candidate hidden units similar to cascade correlation scheme. Therefore, simplicity in hardware implementation is also obtained. Furthermore, this analysis allows us to select from other methods (such as the conjugate gradient descent or the Newton's second order) one of which will be a good candidate for the learning technique. The choice of learning technique depends on the constraints of the problem (e.g., speed, performance, and hardware implementation); one technique may be more suitable than others. Moreover, for a discrete weight space, the theoretical analysis presents the capability of learning with limited weight quantization. Finally, 5- to 8-bit parity and chaotic time series prediction problems are investigated; the simulation results demonstrate that 4-bit or more weight quantization is sufficient for learning neural network using CEP. In addition, it is demonstrated that this technique is able to compensate for less bit weight resolution by incoporating additional hidden units. However, generation result may suffer somewhat with lower bit weight quantization.

I-Introduction

There are many ill-defined problems in pattern recognition, classification, vision, and speech recognition which need to be solved in real time [1-3]. One of the most attractive features of the neural network is a massively parallel processing topology that offers tremendous speed specially when implemented in hardware. Generally, neural network approaches in hardware face two main obstacles:

(1) difficulty of network convergence due to the learning algorithm itself as well as the limited precision of the devices;

(2) high cost of implementing hardware to truly mimic the synapse and neuron transfer functions dictated by the algorithm.

Furthermore, the convergence and the implementable hardware have a mutual correlation to each other; for example, the convergence of the learning network depends on the weight resolution available in synapse [4-6], and the cost of implementation of each bit in synapse grows, at least doubly, in silicon area, power, and connectivity[7-8]

In this paper, CEP learning algorithm is presented. It offers a simple learning method using a one-layer perceptron approach and a deterministic calculation for the other layer. Such a simple procedure offers a fast, reliable, and implementable learning algorithm. In addition, the learning technique is not only tolerant of 3- and 4-bit weight resolution in synapse, but its simplicity indicates the network can be robustly implementable in VLSI hardware. To validate the new learning theory of CEP, simulations for 5- to 8-bit parity and chaotic time series problems are investigated in weight quantization of a floating point machine (32-bit for float and 64-bit for double precision) and are compared with resolutions using synapses with limited weight quantizations (3- to 6-bit weight resolution) of VLSI hardware.

II Mathematical foundation of Cascade Error Projection

1. Continuous weight space:

In this analysis, we only focus on cascading architecture with one hidden unit added one at a time when needed.

Assume that the network contains *n* hidden units (see Fig. 1) and the learning cannot be improved any further in the energy level. At this point, a new hidden unit *n+1* is added to the network.

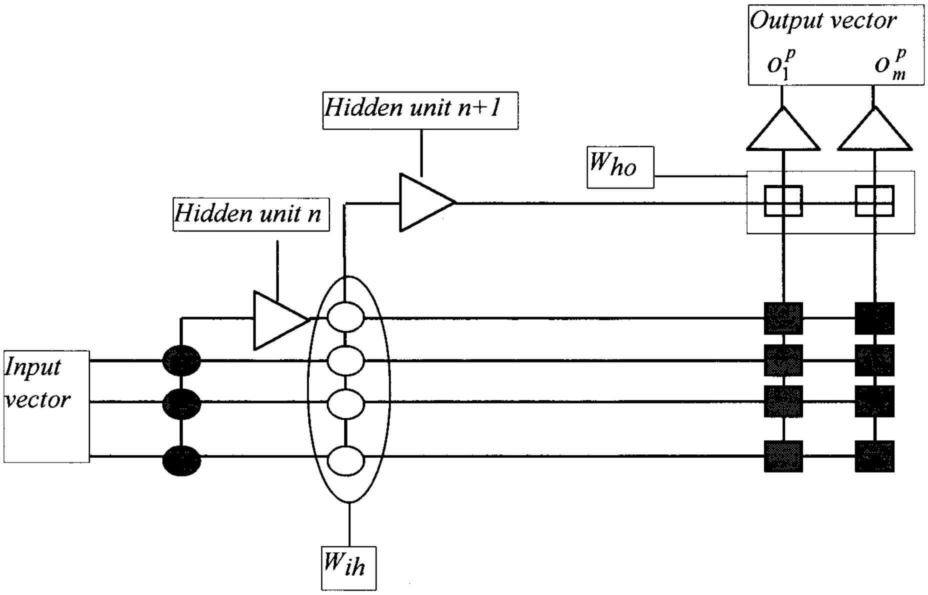


Figure 1: Schematic diagram for CEP learning with a newly added hidden unit (n+1). Blank circles and squares are the weight components that are determined by iterative learning and calculation, respectively.

N is the dimension of the input space, *n+1* is the dimension of the expanded input space (*n+1* is dynamically changed and is based on the learning requirement), and m is the dimension of the output space, P is the number of training patterns. Finally, *f* is a sigmoidal transfer function which is defined by:

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}}$$

Other notations are defined as follows:

W_{ho} denotes the weight vector between newly added hidden unit *n+1* and the output o, and W_{ih} is the weight vector between input units (including original inputs and previous hidden units) and a newly added hidden unit.

$\varepsilon_o^p = t_o^p - o_o^p(n)$ denotes the error for an output index *o* and training pattern *p* between target *t* and the actual output *o(n)*. *n* indicates the output with n hidden units in the network.

$f'^p_o(n)$ denotes the output transfer function derivative with respect to its input index *o* and the training pattern *p*.

$f_h^p(n+1)$ denotes the transfer function of hidden unit *n+1* for a training pattern *p*.

X^p denotes the input pattern *p*.

The energy function is defined as follows:

$$E(i) = \sum_{p=1}^{P} E^p(i) = \sum_{p=1}^{P}\sum_{o=1}^{m}(t_o^p - o_o^p(i))^2 = \sum_{p=1}^{P}\sum_{o=1}^{m}(\varepsilon_o^p)^2$$

The difference of energy between the network with *n* hidden units and the network with *n+1* hidden units can be obtained as,

$$\Delta \mathrm{E} = E(n) - E(n+1) = \sum_{o=1}^{m}\{-w_{ho}^2 \sum_{p=1}^{P}[f'^p_o f_h^p(n+1)]^2 + 2w_{ho}\sum_{p=1}^{P}[\varepsilon_o^p f'^p_o f_h^p(n+1)]\}$$

where $w_{ho} f_h^p(n+1)$ is small. This assumption is needed for nonlinear transformation function only.

As proved in ref. 9, the sufficient condition for maximum energy reduction between hidden unit *n* and a newly added hidden unit *n+1* with respected to w_{ho} is:

$$\max\{(\Delta \mathrm{E})_{Who}\} = \sum_{p=1}^{P}\sum_{o=1}^{m}\{\varepsilon_o^p f'^p_o f_h^p(n+1)\} \quad \text{when } w_{ho} = \frac{\sum_{p=1}^{P} \varepsilon_o^p f'^p_o f_h^p(n+1)}{\sum_{p=1}^{P}[f'^p_o f_h^p(n+1)]^2} \qquad (1)$$

Let

$$\Gamma = \begin{bmatrix} \frac{1}{m}\sum_{o=1}^{m} f'^1_o\{t_o^1 - o_o^1\} \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \frac{1}{m}\sum_{o=1}^{m} f'^P_o\{t_o^P - o_o^P\} \end{bmatrix}$$

Then, $\Gamma \in [-1,1]^P$.

and

$$F_h(n+1) = \begin{bmatrix} f_h^1(n+1) \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ f_h^P(n+1) \end{bmatrix}$$

We can rewrite equation (1) using a matrix notation as follows:

$$\Delta \mathrm{E} = m\Gamma^{T} F_{h}(n+1) \qquad (2)$$

From (1) and (2), the energy reduction is dependent on a match between Γ and $F_{h}(n+1)$. The technique to match Γ with $F_{h}(n+1)$ can include, e.g. perceptron learning with gradient descent, maximum correlation or covariance with gradient ascent, conjugate gradient, and Newton's second order method. Therefore, the learning network performance really depends on the learning technique chosen for matching the error surface Γ and $F_{h}(n+1)$. In equation (2), let $f'^{p}_{o}(n)=1$; then it can be rewritten as:

$$\Delta \mathrm{E} = \sum_{p=1}^{P} \{ f_{h}^{p}(n+1) \frac{1}{m} \sum_{o=1}^{m} (t_{o}^{p} - o_{o}^{p}(n)) \} \qquad (3)$$

Thus equation (3) is a special case for the general formulation of equation (2). From (3), maximum correlation or covariance is applied to maximize ΔE, then it represents the technique of cascade correlation learning algorithm which has been reported in literature by Fahlman [10].

2. Discrete Weight Space:

In continuous weight space, the weight quantization can be considered as infinite. However, in hardware, weight quantization is always finite and limited. Therefore, it is necessary to convert the weight updates Δw to a finite weight quantization Δw*. As proved in ref. 9, learning can be done with limited weight quantization as long as the difference between Δw and Δw* is viewed as equivalent independent white noise (round-off conversion technique) and the stepsize which is used to convert from Δw to Δw* must not be fixed. The dynamical stepsize can be roughly estimated as follows:
In continuous space, the energy reduction is:

$$\Delta \mathrm{E} = \sum_{p=1}^{P} \sum_{o=1}^{m} \varepsilon_{o}^{p} f'^{p}_{o} f_{h}^{p}(n+1)$$

During learning, limited weight quantization value, Δw* directly affects the output of the (n+1)th hidden unit $f_{h}(n+1)$. It is expressed as:

$$\tilde{f}_{h}^{p}(n+1) = f_{h}^{p}\left(\sum_{i=1}^{N+1} \tilde{w}x_{i} + \sum_{j=1}^{n} \tilde{w}x_{h}(j)\right),$$ $\tilde{w}$ is a weight component in finite weight space.

The reduction of energy in discrete (finite) weight space [9] is:

$$\Delta\tilde{\mathrm{E}} = \tilde{\mathrm{E}}(n) - \tilde{\mathrm{E}}(n+1) = \sum_{p=1}^{P} \sum_{o=1}^{m} \varepsilon_{o}^{p} f'^{p}_{o} \tilde{f}_{h}^{p}(n+1) \qquad (5)$$

Our main focus is finding the conversion factor (stepsize) which is based on the known factor (e.g. previous energy E(n)), thereby the conversion factor (stepsize) can be estimated, and the learning being conducted in limited weight quantization, can be enhanced.

$$\Delta\tilde{\mathrm{E}} \propto \tilde{f}_{h}^{p}(n+1) \qquad (6)$$

Ignoring the nonlinear characteristic, it is roughtly estimated that:

$$\tilde{f}_{h}^{p}(n+1) \propto \tilde{w} \propto stepsize(n+1) \qquad (7)$$

From (5), (6), and (7) one can express

$$stepsize(n+1) \propto \widetilde{\mathrm{E}}(n) \qquad (8)$$

The expression in (8) is a critical step in estimating the dynamical stepsize which is dependent on the previous energy of the network. In other words, the expression can be written as:

$$stepsize(n+1) = \alpha\widetilde{\mathrm{E}}(n)$$

The value of α was obtained ad hoc for each application through experiments.

III. Simulations Using CEP

As reported in [11-12], we use gradient descent technique for learning W_{ih} *and* calculate the W_{ho}.

1) Problems:

Using this technique, we have solved:

- 5- to 8-bit parity problems
- chaotic time series problem

The constraints for weight space are:

a. no limited weight quantization (floating point 32-bit for single precision and 64-bit for double precision); and,

b. the limited weight quantization from 3- to 6-bit for parity and 4- to 6-bit for chaotic time series problems.

2) Conversion technique (round-off technique)

$$\Delta w^{*}_{jh}(n) = \begin{cases} stepsize(n) * \mathrm{int}(\dfrac{\Delta w_{jh}}{stepsize(n)} + 0.5) & \textit{if } (\dfrac{w_{jh}(n)}{stepsize(n)} + \mathrm{int}(\dfrac{\Delta w_{jh}(n)}{stepsize(n)} + 0.5)) \le 2^{B} \textit{ and } \Delta w_{jh}(n) > 0 \\ stepsize(n) * \mathrm{int}(\dfrac{\Delta w_{jh}(n)}{stepsize(n)} - 0.5) & \textit{if } (\dfrac{w_{jh}(n)}{stepsize(n)} + \mathrm{int}(\dfrac{\Delta w_{jh}(n)}{stepsize(n)} - 0.5)) \le -2^{B} \textit{ and } \Delta w_{jh}(n) < 0 \\ 0 & \textit{Otherwise} \end{cases}$$

3) Simulation results:

Parity Problem:

As noted earlier, we are solving 5-,6-,7-, and 8-bit parity with different synaptic resolution. We compare the results of higher and lower synaptic resolution to show the robustness of such an algorithm for hardware implementation [11-12].

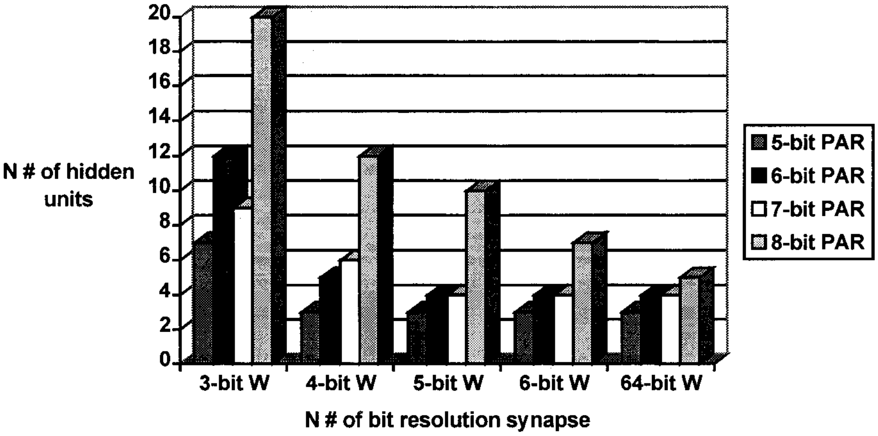
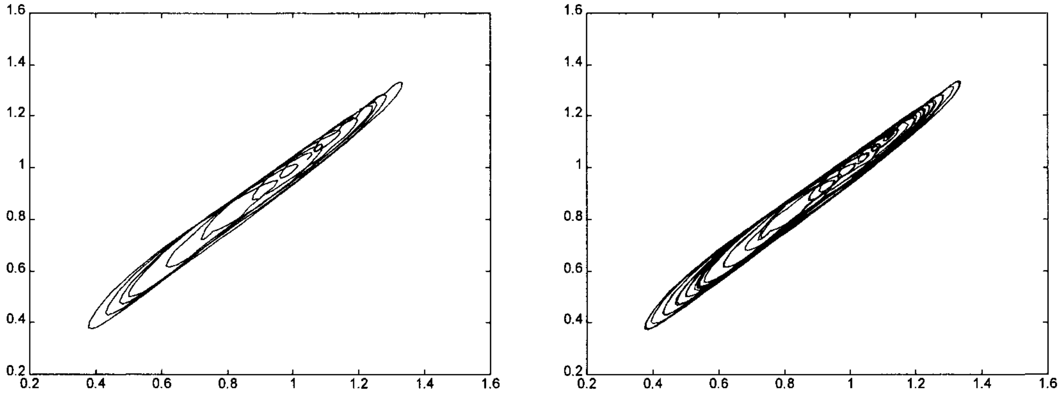


Figure 2: The chart shows CEP learning capability and the number of hidden units required to correctly solve 5- to 8-bit parity problems using round-off technique. x axis represents weight quantization (3-6 and 64-bit) and y axis shows the resulting number of hidden units (limited to 20). Each learning hidden unit is provided with 100 epoch iterations. As shown, a lager number of hidden units compensate for the lower weight resolution.

Chaotic Time Series Problem:

The data in this problem represents chaos and never repeated. However, this data between past, present, and future are correlated in high order. To validate the capability of CEP as shown in theory, we use CEP learning technique under constraints of limited weight quantization (4-, 6-, and 64-bit weight resolution) to capture the high order correlation of this problem.

In this experiment, we use x_i, x_{i+1}, x_{i+2}, x_{i+3} and the target is x_{i+4} . The number of training data is 351 and test data is 651 and no cross validating data is applied in this phase.



(a) (b)

Figure 3: Data sets of chaotic time series problem. (a). training set to the CEP neural network, and (b). Test set which has no overlap with training set.

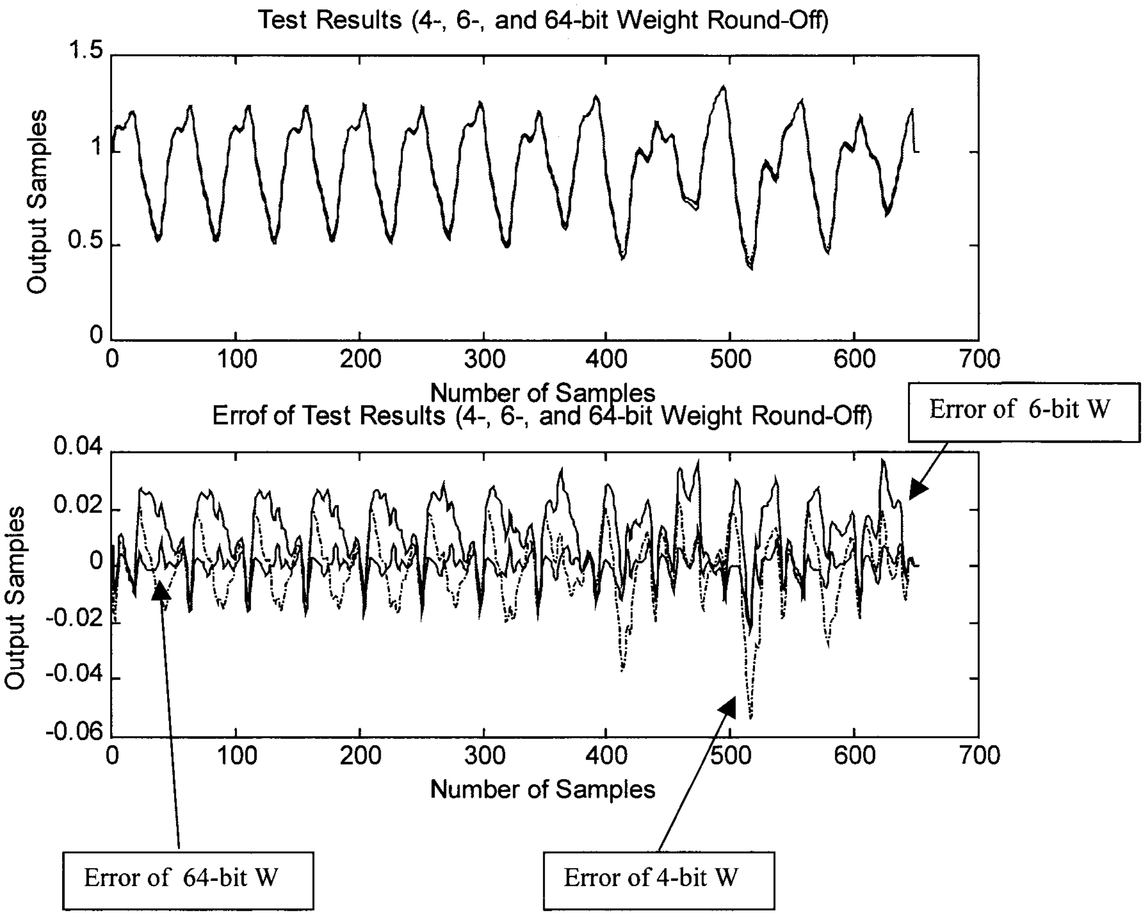


Figure 4: Simulation Results of CEP for chaotic time series prediction problem. Top trace contains four curves: ideal data, 64-bit, 6-bit and 4-bit prediction results. Bottom trace contains : errors between ideal data and 64-bit, 6-bit, and 4-bit generalization data.

The results in Figure 4 show that the error between ideal data and prediction with 64-bit weight learning network is within +/-0.01 and is like white noise, whereas, 6-bit error is more harmonic than 4-bit error prediction. These results can be interpreted to infer that the more bit weight quantization is available for learning the better and smoother the transform would be. In addition, the better and smoother transformation will help network to interpolate for predictions.

IV. Conclusions

In this paper, we have shown that CEP is a reliable technique for both software- and hardware-based neural network learning. From this analysis, it is shown that the CC algorithm is a special case and can be understood in greater depth with this analysis. Moreover, the theoretical analysis provides us with the general framework of the learning architecture, and the particular learning algorithm can be independently studied for its suitability for a given application associated with given constraints specific to each problem. For example, for hardware implementation CEP is advantageous, but for software, covariance or Newton's second order method is more advantageous). For the CEP learning algorithm, the advantages can be summarized as follows:

- A fast and reliable learning technique
- A hardware implementable learning technique
- Learning scheme is tolerant of lower weight resolutions.
- A robust model in learning neural networks

Acknowledgments:

The research described herein was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was jointly sponsored by the Ballistic Missile Defense Organization/Innovative Science and Technology Office (BMDO/IST), and the National Aeronautics and Space Administration (NASA). The authors would like to thank Drs A. Stubberud and A. Thakoor for useful discussions.

References:

[1] T. A. Duong, T. Brown, M. Tran, H. Langenbacher, and T. Daud, "Analog VLSI neural network building block chips for hardware-in-the-loop learning," *Proc. IEEE/INNS Int'l Join Conf. on Neural Networks*, Beijing, China, Nov. 3-6, 1992.

[2] T. A. Duong et. al, "Low Power Analog Neurosynapse Chips for a 3-D "Sugarcube" Neuroprocessor," *Proc. of IEEE Intl' Conf. on Neural Networks*(ICNN/WCCI), Vol III, pp. 1907-1911, June 28-July 2, 1994, Orlando, Florida.

[3] B.E. Boser, E. Sackinger, J. Bromley, Y. LeCun, and L.D. Jackel, "An Analog Neural Network Processor with Programmable Topology," *IEEE Journal of Solid State Circuits*, vol. 26, NO. 12, Dec. 1991.

[4] P. W. Hollis, J.S. Harper, and J.J. Paulos, "The effects of Precision Constraints in a Backpropagation learning Network," *Neural Computation*, vol. 2, pp. 363-373, 1990.

[5] M. Hoehfeld and S. Fahlman, "Learning with limited numerical precision using the cascade-correlation algorithm," *IEEE Trans. Neural Networks*, vol.3, No. 4, pp 602-611, July 1992.

[6] T.A. Duong, S.P. Eberhardt, T. Daud, and A. Thakoor, "Learning in neural networks: VLSI implementation strategies," In: *Fuzzy logic and Neural Network Handbook*, Chap. 27, Ed: C.H. Chen, McGraw-Hill, 1996.

[7] S.P. Eberhardt, T.A. Duong, and A.P. Thakoor, "Design of parallel hardware neural network systems from custom analog VLSI "building-block" chips," *IEEE/INNS Proc. IJCNN*, June 18-22, 1989 Washington D.C., vol. II, pp. 183.

[8] T. A. Duong, S. P. Eberhardt, M. D. Tran, T. Daud, and A. P. Thakoor, "Learning and Optimization with Cascaded VLSI Neural network Building-Block Chips," *Proc. IEEE/INNS International Join Conference on Neural Networks*, June 7-11,1992, Baltimore, MD, vol. I, pp. 184-189.

[9] T. A. Duong, *Cascade Error Projection_An sufficient Hardware learning theory. Ph.D. Thesis, UCI*, 1995.

[10] S. E. Fahlmann, C. Lebiere, "The Cascade Correlation learning architecture," in *Advances in Neural Information Processing Systems II*, Ed: D. Touretzky, Morgan Kaufmann, San Mateo, CA, 1990, pp. 524-532.

[11] T.A. Duong, "*Cascade Error Projection-An efficient hardware learning algorithm*," Proceeding Int'l IEEE/ICNN in Perth, Western Australia, vol. 1, pp. 175-178, Oct. 27-Dec 1, 1995 *(Invited Paper).*

[12] T.A. Duong, A. Stubberud, T. Daud, and A. Thakoor, "*Cascade Error Projection-A New Learning Algorithm,*" Proceeding Int'l IEEE/ICNN in Washington D.C., vol. 1, pp. 229-234, Jun. 3-Jun 7, 1996.